Lab 6 - Code Design 2

Sunday, April 12, 2020 12:14 PM

Notes: On the LCD display - first line is 0, second is 1. First column is 0.

```
If(UART has input char)
        Increment char_count
        If (printable character)
                Output char to LCD display
        If((line=1) AND ((input=CR) OR (char_count=16))
                line=2
        else
                if ((line=2) and ((input is CR) OR (char_count=16))
                        Copy line 2 content to line 1 buffer
                        If (char_count<16) pad buffer with spaces
```

Move LCD cursor to 1,1		// start of line 1

Write buffer 1 content to LCD

Move LCD cursor to 2,1		// start of line 2
char_count = 0;		// start again on line 2

```
#include <string.h>
#define BUF_LEN 17            // input buffer lengths

   char buffer1[BUF_LEN];        // for scrolling messages
   char buffer2[BUF_LEN];
```

```
                        // save chars in buffer
      buffer2[char_count-1] = c;    // write to buffer - watch c array indicies 0..n-1
      if((line==2) && ((c==13)||(char_count==16))) {
         for (i=char_count;i<=BUF_LEN;i++)
           buffer2[i] = 32;                // fill buffer with spaces
      if (c==13)
           buffer2[char_count-1] = 32;       // compensate for CR
```

```
    strncpy(buffer1, buffer2,BUF_LEN);    // move buffer2 chars to buffer1

     /* ----------- Display line 2 chars on line 1 ------------ */
      /*------- LCDS cusor command sequence --------*/
     SPI1BUF=0x1b;                        // Cursor move - first send escape char
     c_buffer = "[j";                     // command sequence for clear display and home cursor
     putsSPI1(2,c_buffer);               // write out string
     DelayMs(500);                        // wait for display to reset

     putsSPI1(BUF_LEN,buffer1);          // write out string
     DelayMs(500);                        // wait for display to reset

     /* -------------- start a new line 2 ---------------*/
     /*------- LCDS cusor command sequence --------*/
     SPI1BUF=0x1b;                       // Cursor move - first send escape char
     c_buffer = "[1;0H";                      // command sequence for cursor move
     putsSPI1(5,c_buffer);       // write out string
     DelayMs(500);                       // wait for display to reset

     char_count = 0;      // reset for new line
}
```